A Project Report on

*i*Bot -

A WIRELESSLY CONTROLLED VIDEO SURVEILLANCE ROBOT

Under the Guidance of

Prof. Vasanth Kumar,
Asst. Professor, Dept. of E&C.

Project Associates

Roger Julian Pinto (4NM01EC079)
Roshan Lawrence Valder (4NM01EC081)
Sanath S. Kumar (4NM01EC085)
Vikram Radhakrishna Nayak (4NM01EC119)

Project Report submitted to the Visveswaraiah Technological University, in partial fulfillment of the requirements for the award of Bachelor of Engineering in Electronics and Communication Engineering.

Department of Electronics and Communication Engineering,
NMAMIT, Nitte.

NITTE MAHALINGA ADYANTHAYA MEMORIAL INSTITUTE OF TECHNOLOGY

NITTE – 574110.

DEPARTMENT OF ELECTRONICS AND COMMUNICATION

CERTIFICATE

Certified that the project work entitled **iBot - A Wirelessly Controlled Video Surveillance Robot** is a bona fide work carried out by **Roger Julian Pinto, Roshan Lawrence Valder, Sanath S. Kumar** and **Vikram Radhakrishna Nayak** in partial fulfillment of the requirements of the Visveswaraiah Technological University, Belgaum during the year **2005**. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the Bachelor of Engineering Degree.

Signature of Guide Signature of HOD Signature of Principal

Name of the Student:

University Seat Number:

External Viva

Name of the examiners Signature with date

1.

2.

ABSTRACT

***i*Bot** is a highly maneuverable wirelessly controlled video surveillance robot. The robot has a wireless camera mounted on it to display live video to the user, using a Graphical User Interface (GUI) on a computer. The GUI is also used to wirelessly control the robot. The bot can be moved Front/Back/Left/Right and has a 0° turning radius. Tilt Up/Down controls are available for the camera. Touch sensors are present on the robot so that feedback is available and the bot detects obstacles in its path.

ACKNOWLEDGEMENTS

We would like to express our sincere thanks to our guide **Prof. Vasanth Kumar**, Asst. Professor, Department of Electronics and Communication, NMAMIT, for guidance and invaluable help in developing our project.

We would like to thank our H.O.D. **Dr. M. K. Parasuram** for letting us avail the laboratory facilities and for his timely inputs.

We would also like thank our respected principal, **Dr. S. Y. Kulkarni** for the encouragement provided to us in undertaking this project.

We would like to express our gratitude to all those who have directly or indirectly helped us in this endeavour, including the service and maintenance faculty in the E&C department.

Also, a special thanks to all our friends and well wishers who have helped us in the completion of this project.

- The iBot Project Team

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

Robots are used in all aspects of life nowadays. Robots are used in industry for applications such as assembly lines, precision machining, spot welding, painting, machine loading, parts transfer, etc. Robots are used in environments that pose great danger to humans, such as locating sunken ships, cleanup of nuclear waste, prospecting for underwater mineral deposits, and active volcano exploration. Robots are also used in the home environment; examples are Roomba, the vacuuming robot and Aibo, the robotic dog. Robots are able to perform repetitive tasks more quickly, cheaply, and accurately than humans.

The word ‘robot’ comes from the Czech word *robota* meaning ‘forced labour’. A robot is defined as a computer-controlled machine that is programmed to move, manipulate objects, and accomplish work while interacting with its environment.

In the modern society, great emphasis is laid on security. For any business or even a home, a security device has become indispensable. Hence, this project combines the aim of ‘security’, with the concept of robotics.

Our project, *iBot*, is a wirelessly controlled video surveillance robot. It is designed to be a complete surveillance device that enhances the security capabilities of any organization or home. It will help a user to set up an efficient surveillance at his home or work place and monitor the activities that happen from his computer screen.

Chapter 2

OBJECTIVES AND DESIGN METHODOLOGY

2.1 Project Objectives

We started working on our project with the following objectives:

- Project must serve as an effective surveillance device
- Wireless control by the user through a computer
- Live full motion video displayed to the user on the monitor
- Mechanical design to be so as to ensure high maneuverability for the robot
- Pan and Tilt mechanism for the camera
- Modular design (Ability to implement and test in stages)
- Sensors to be implemented on the robot to provide feedback
- If time allows, add a degree of artificial intelligence to the robot

2.2 Design Methodology

One of our design objectives was to have a modular design i.e. the ability to implement and test in stages and we have followed this methodology throughout the project. Each individual module was independently built and tested. The designs were made such that it could directly be connected to the next module, and changes/improvements could be made in preceding or subsequent modules without affecting the inputs/outputs of the current module.

To provide effective surveillance, the requirement was that the device must be mobile and the control unit must be stationary. A computer was to be used as the control unit, as it was very user friendly, and also provided a large number of options for interfacing. It was decided to use the Parallel port for interfacing the circuitry to the computer. The other options considered were the RS-232 & RS-485 Serial Ports, and the USB port, which are faster than the parallel port. But since only control signals were being sent, the parallel port was determined to be sufficient. Since parallel ports in general do not have much in built protection circuitry, it was necessary to use opto couplers to interface it to the control circuit.

For the wireless control, it was decided to use DTMF (Dual Tone Multi Frequency) signals transmitted over FM (Frequency Modulation). DTMF was chosen because of the

ease at which signal could be transmitted. For DTMF generation, IC chosen was UM91214B which can generate 12 DTMF signals. This was sufficient since only 6 distinct control signals (Front/Back/Left/Right and Tilt Up/Down) needed to be transmitted. There was scope for future enhancements using upto 6 different signals with only minor circuit modifications. The DTMF generator circuit was interfaced to the parallel port using opto couplers and CMOS switches. The MT 8870 IC was used to decode the DTMF signals. The output of the MT 8870 is in binary; hence a combinational circuit was designed using a multiplexer 74154 and 7400 NAND gates to generate distinct control signals to be given to the motor driver circuitry.

For FM transmission and reception, widely available readymade FM transmitter and receiver boards were used, because of the difficulty involved in fabricating them. The 1.2GHz wireless camera with receiver was also purchased separately. The video output from the camera was given to a TV Tuner card on the computer to display the video to the user via the GUI.

For the user interface, Microsoft Visual Basic was used, because of its ease of learning and programming. The platform used was Windows XP, which does not allow direct port access. Hence, it was required to use a DLL (Dynamic Link Library) file called `inpout32.dll` downloaded from the internet which allowed us to use the parallel port even under Windows XP.

The robot was designed to use two driving wheels connected to the two motors using belt drives. To move forward or backward, both motors would turn in the same direction, and to turn, the motors would move in opposite directions. This ensured high maneuverability and a turning radius of zero. For the motion, we chose DC motors because the control circuitry (popularly known as an H-Bridge) is easier to implement as compared to stepper motors. The DC Motors used for the project were salvaged from discarded CD drives. A third motor for the camera tilt mechanism was taken from an old VCR.

To provide feedback from the robot to the computer multiple touch sensors were used. The feedback modules worked on the same concept as transmission of control signals i.e. it was achieved through DTMF over FM. To send all possible combinations of the states of all the sensors, a combinational circuit was designed using a 3:8 decoder IC 74138 and NOT gates (IC 7404), and the outputs were used to enable electronic switches (IC 4066) to short the required pins on the DTMF encoder IC 91214B.

Chapter 3

BLOCK DIAGRAM AND SCHEMATICS

3.1 Block Diagram

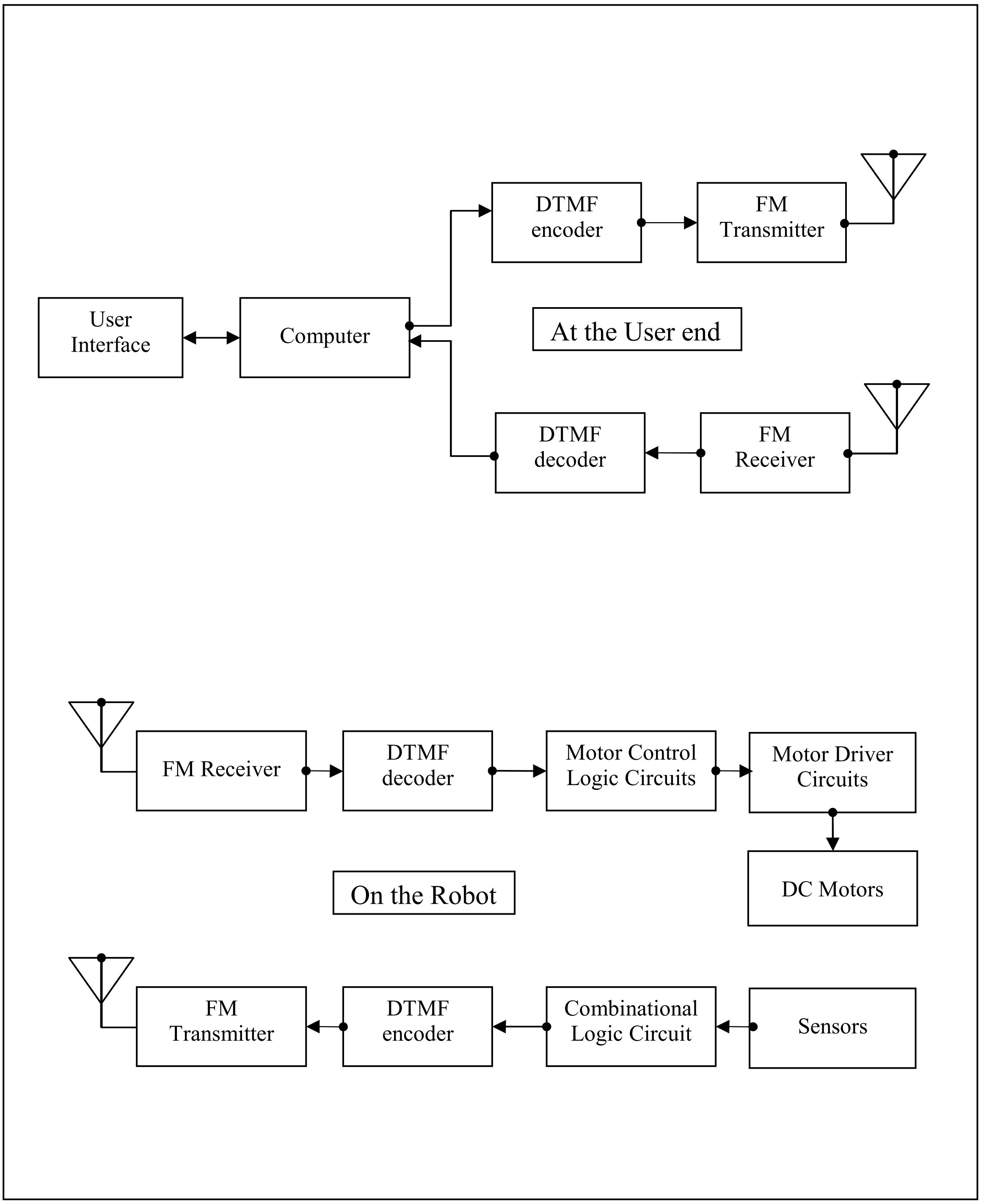


Fig. 3.1: Block diagram of *iBot*

3.2 Circuit Diagrams

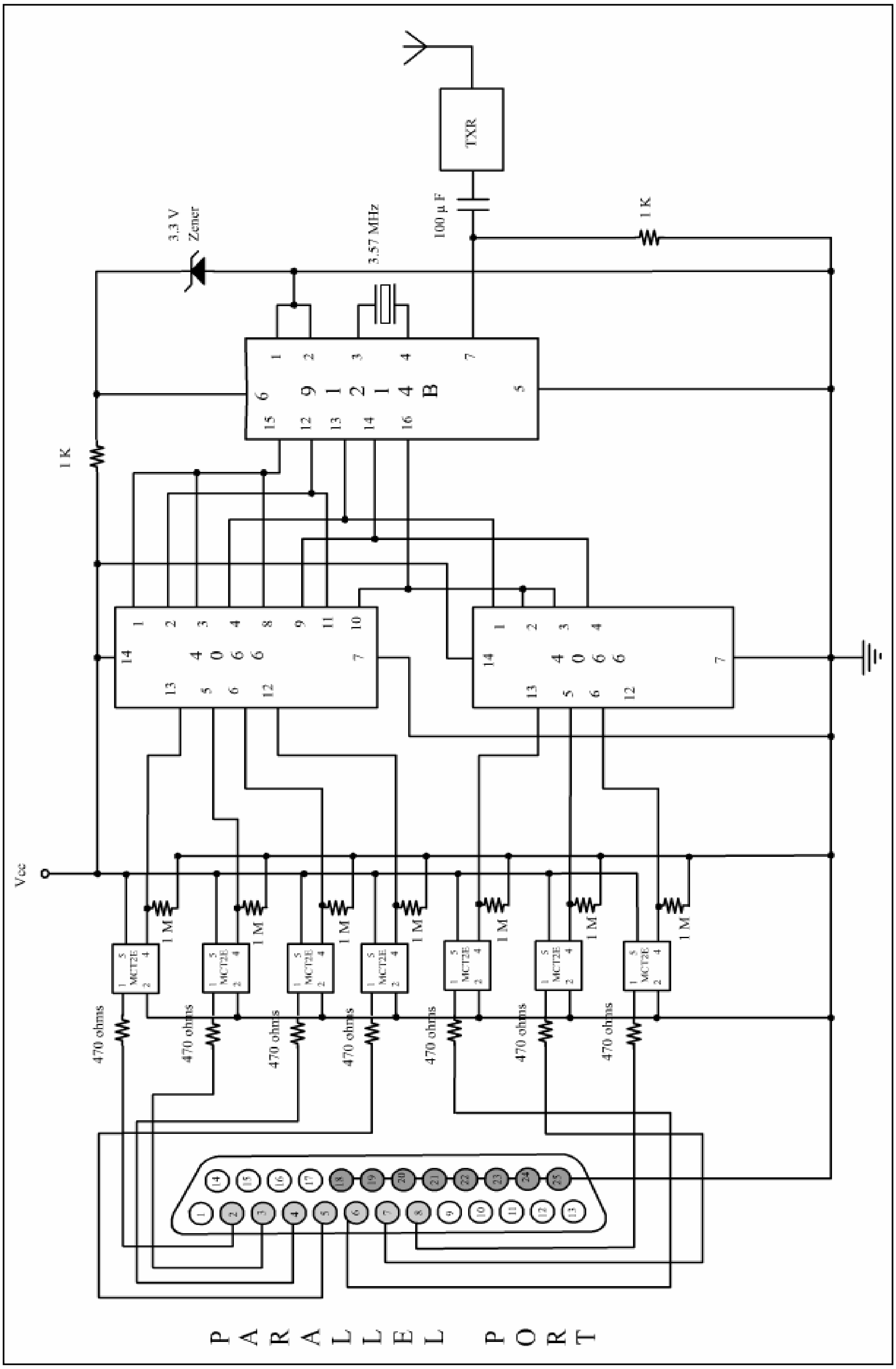


Fig. 3.2: Parallel Port Interfacing and DTMF Encoder Circuit

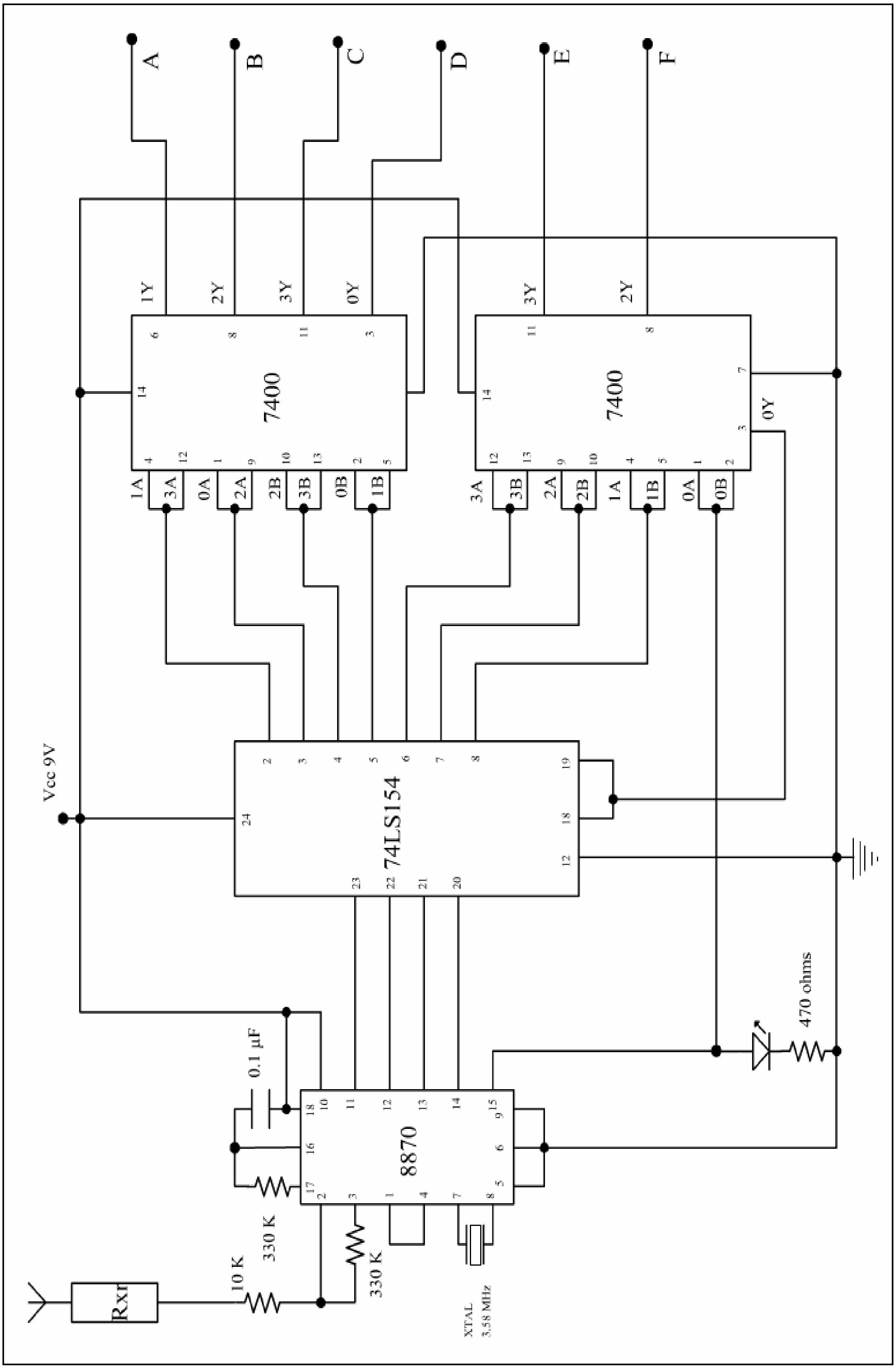


Fig. 3.3: DTMF Decoder and Motor Controller Logic Circuit

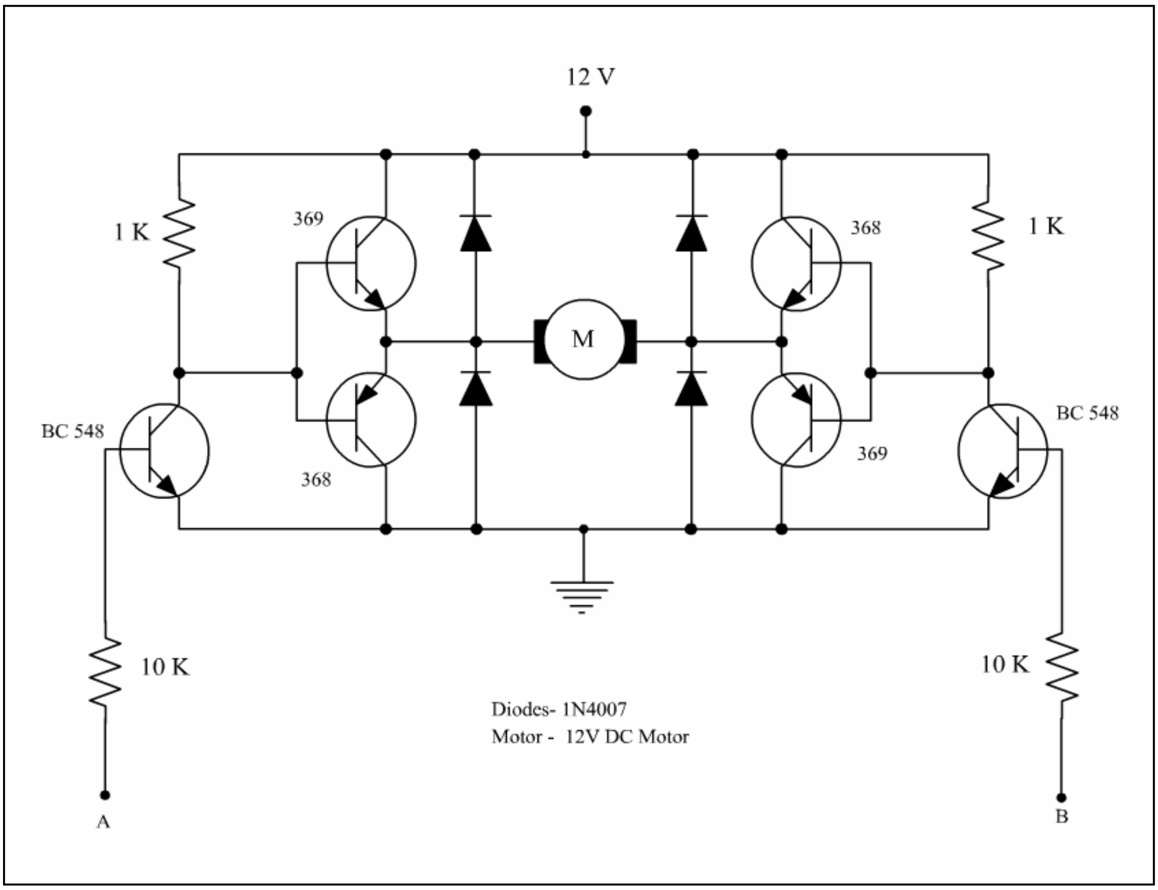


Fig. 3.4: H-Bridge Motor Driver Circuit

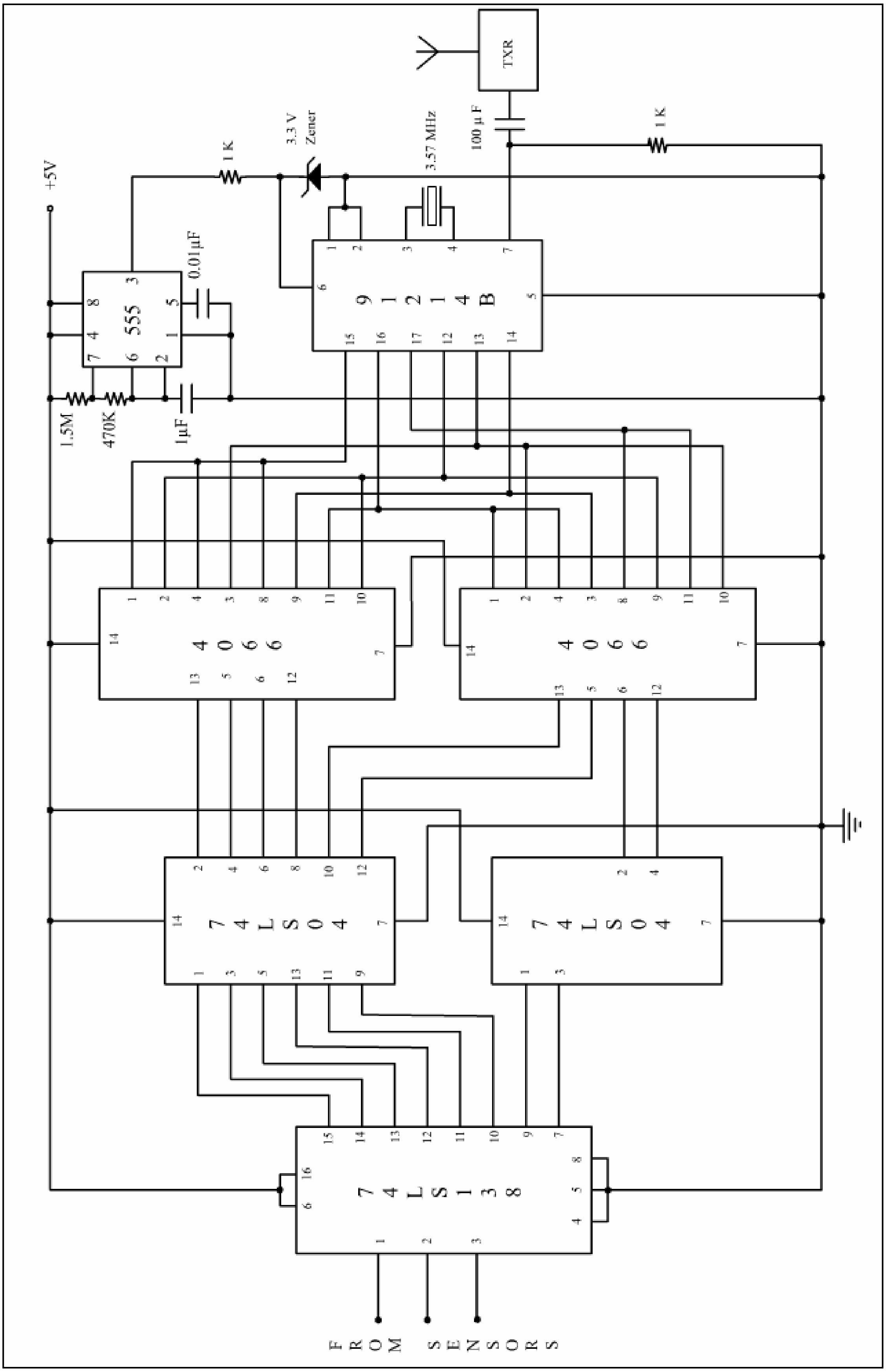


Fig. 3.5: Feedback Module - Transmitter Section

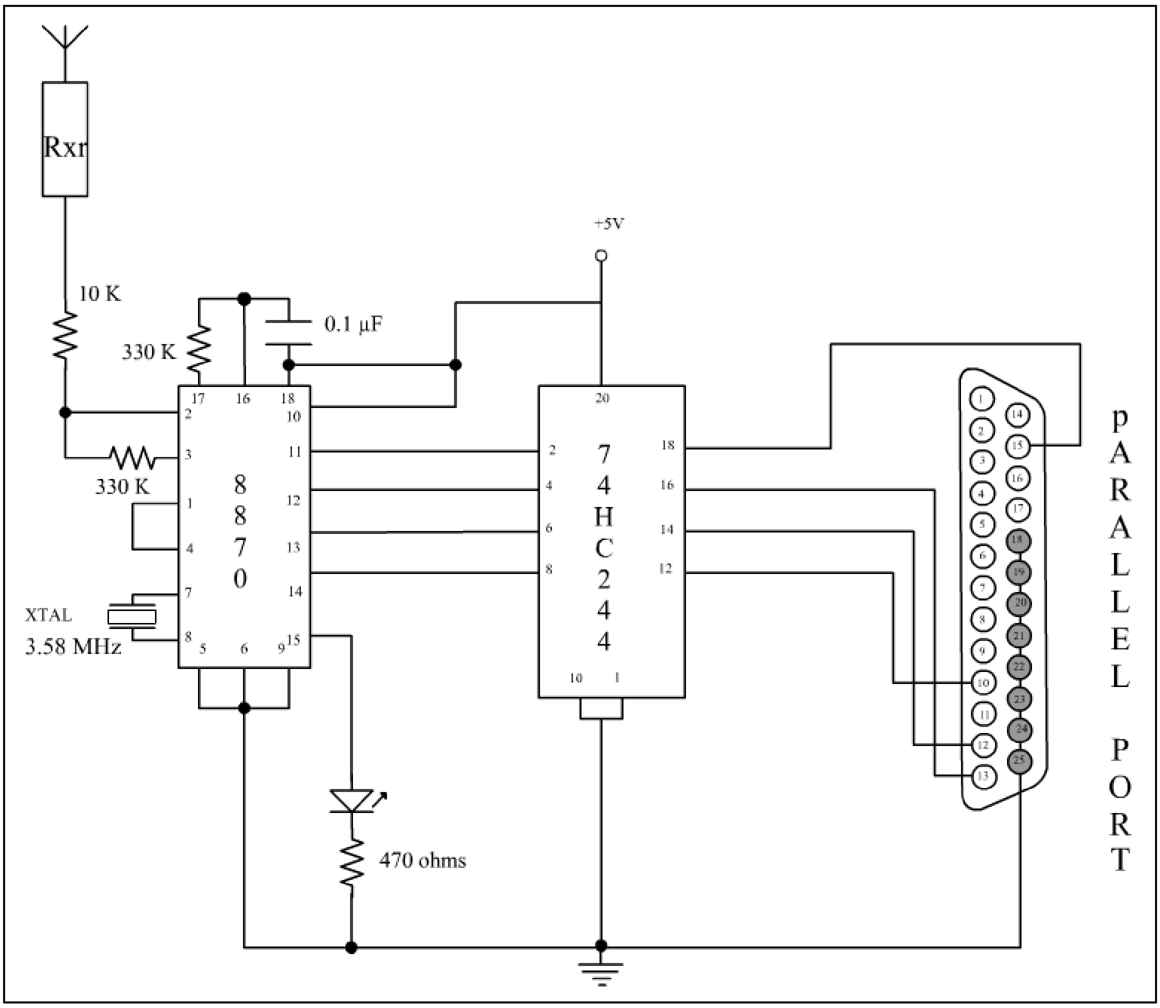


Fig. 3.6: Feedback Module - Receiver Section

Chapter 4

DETAILED EXPLANATION

A main feature of our project is its modular design. The entire project was assembled module by module. This enabled individual testing and debugging of modules before arriving at the final project. Thus, *iBot* is comprised of many smaller modules and not just a single piece of circuitry.

The modules can be largely classified under two fields:

- Hardware, and
- Software.

The software is implemented on a PC using Microsoft Visual Basic 6. The hardware is mainly composed of digital ICs. This section will discuss in detail the hardware design and the software.

4.1 Parallel Port Interfacing

To achieve control of the robot through computer, the parallel port of the PC is used. Pins 2 to 9 on the parallel port are the data pins D0-D7. These pins are used for sending the control signals from the computer to the robot. MCT2E optocouplers are used to protect the parallel port i.e. isolate the current of the circuit and current of the parallel port. Each of the 8 output data bits from the parallel port of the PC are given to pin 1 of the optoisolator through a 470Ω resistor to minimize the source current from the parallel port. A signal on pin 1 switches on the internal infra red LED, and the emitted light falls on the photo transistor, hence turning it ON, and the output is available at pin 4.

4.2 DTMF Encoder

For transmitting control signals, DTMF (Dual Tone Multi Frequency) tones are used. The DTMF tones are generated using the IC UM91214B. This IC needs only an inexpensive 3.57MHz crystal to generate the DTMF tones. If a matrix similar to a telephone keypad is imagined, a DTMF tone corresponding to a number is generated by the IC when the

corresponding Row and Column pins are shorted. To short the required pins depending on the outputs from the parallel port, the IC 4066 CMOS switches are used.

The outputs of the optocouplers are given to the "enable" pins of the IC 4066 CMOS switches. This IC completes the circuit at the output pins when a high signal is present at the corresponding enable pin. The outputs are connected to the required Row and Column pins of the UM91214B DTMF generator IC. Thus a unique DTMF tone corresponding to the key pressed by the user is obtained at the output of the DTMF generator.

4.3 Wireless Transmission and Reception

To achieve wireless transmission of the control signals to the robot, the method used is Frequency Modulation (FM). For FM transmission, ready made "Wireless Mic" modules are used. At receiving end, on the robot, we have the Wireless Mic receiver. The receiver output is amplified by a TBA810 Audio Amplifier Board, and fed to the input of the DTMF Decoder.

4.4 DTMF Decoding

For DTMF decoding, the IC used is the MT8870. This IC can decode all 16 DTMF tones and the output is a 4 bit binary, corresponding to the digit received. To get individual mutually-exclusive outputs, the 4 bit binary is given to a 4:16 Demultiplexer, IC 74154 (which has active low outputs). These outputs are then used to control the motors using the designed motor controller circuitry.

4.5 Motor Driving Circuit

For driving the motors, the circuit used is the classic H-Bridge circuit (Fig 3.4), which enables us to start/stop the motor, as well as control its direction. The circuit has two control inputs 'A' and 'B', and depending on the logic status of the control inputs, the motor state and direction is determined.

4.6 Motor Controller Logic Circuit

Our design uses 3 motors - 2 motors for the driving wheels and one for the camera tilt mechanism. We are using DTMF for control and an H-Bridge circuit for driving the motors. Hence an appropriate logic circuit was designed to decode the DTMF signals received and control the state and directions of the required motor(s).

The motor driver logic circuitry requirement is given below.

DTMF Digit	Robot Action	Motor 1	Motor 2	Motor 3
1	Move Forward	ON-F	ON-F	OFF
2	Move Reverse	ON-R	ON-R	OFF
3	Turn Left	ON-R	ON-F	OFF
4	Turn Right	ON-F	ON-R	OFF
5	Camera Up	OFF	OFF	ON-F
6	Camera Down	OFF	OFF	ON-R

Table 4.1: Motor driver logic circuitry requirement

Inputs (from Demultiplexer)						Outputs (To H-Bridges)					
						Motor 1		Motor 2		Motor 3	
$\overline{D1}$	$\overline{D2}$	$\overline{D3}$	$\overline{D4}$	$\overline{D5}$	$\overline{D6}$	A	B	C	D	E	F
1	0	0	0	0	0	1	0	1	0	0	0
0	1	0	0	0	0	0	1	0	1	0	0
0	0	1	0	0	0	0	1	1	0	0	0
0	0	0	1	0	0	1	0	0	1	0	0
0	0	0	0	1	0	0	0	0	0	1	0
0	0	0	0	0	1	0	0	0	0	0	1

Table 4.2: Motor driver logic circuitry truth table

Solving for A, B, C, D, E and F,

$$A = \overline{D1} + \overline{D4} = \overline{D1.D4} \qquad \ldots(4.1)$$

$$B = \overline{D2} + \overline{D3} = \overline{D2.D3} \qquad \ldots(4.2)$$

$$C = \overline{D1} + \overline{D3} = \overline{D1.D3} \qquad \ldots(4.3)$$

$$D = \overline{D2} + \overline{D4} = \overline{D2.D4} \qquad \ldots(4.4)$$

$$E = \overline{D5} \qquad \ldots(4.5)$$

$$F = \overline{D6} \qquad \ldots(4.6)$$

These equations are implemented using NAND gates (IC 7400) as is seen in the circuit diagram (Fig 3.3).

4.7 Mechanical Construction

For the mechanical construction, our main aim was high maneuverability. Hence, we have used a differential drive system with two driven wheels and one free wheel for balance. Thus, our robot is highly maneuverable and has a turning radius of 0°. The two driving motors are DC motors. A third DC motor is used for the Camera Tilt assembly.

4.8 Wireless Video Transmission

A ready made wireless camera module is used. The camera has an in built 1.2GHz transmitter and comes with its own RF receiver. The output from the receiver is composite video which can be viewed on a TV. Since we are using a computer, a TV tuner card was necessary to view the analog signals. Using the software, the live video from the camera is displayed to the user in the GUI.

4.9 Feedback

A secondary objective of our project was to have feedback available from the robot to the computer. For this purpose, the status pins of the parallel port are used. Touch sensors are mounted on the robot. The three signals obtained from the 3 sensors are given as inputs to a 3:8 decoder (IC 74138). The eight output lines are given to IC 4066 CMOS switches to short two pins on the UM91214B IC to generate unique DTMF tone outputs. Wireless transmission is achieved using another FM transmitter/receiver pair. At the computer end, an MT8870 IC decodes the received DTMF into a 4-bit binary. This is given to the status port of the computer through a 74LS244 buffer IC. It is decoded in software and required action is taken.

4.10 Software

Visual Basic was used as the programming language because of its ease of learning and implementation. Since the operating system used was Windows XP, and it did not allow

direct port access, a DLL (Dynamic Link Library) known as `inpout32.dll` was used to access the parallel port. This DLL has two functions *inp* and *out*, which correspond to reading data and writing data to the ports.

The program has many modules and a single form that accesses these modules. The form which contains buttons and a picture window is the front end through which the user communicates to the robot, giving it commands and viewing video. The code that is contained in the form sends data to the data port when the appropriate movement button is pressed. The code for repetition of a sequence of motion is also provided here.

Basic routines are written in the form of modules which the code contained in the form can access. This includes various reads and writes to different ports, video manipulation routines, API declarations, constants etc.

Code related to video streaming can be found on the form and under modules VBAVICAP.BAS and VBMEMCAP.BAS. These modules contain API declaration, basic routines etc. that perform video manipulation. The code is derived from public domain code by E. J. Bantz, available online at http://ej.bantz.com.

The Graphical User Interface has the following components:

- Control Buttons: They enable the user to control the movement of the robot and the on-board camera.
- Video Window: displays live video from the robot camera to the user.
- Parallel Port monitor: displays the current states of the Data and Status pins of the parallel port.
- Patrol Mode Options: Enable the user to program a sequence of operations for the robot to continuously carry out.
- Feedback Monitor: Displays to the user the current state of the sensors.

Chapter 5

COMPONENTS AND DESCRIPTIONS

5.1 DTMF (Dual Tone Multi Frequency)

DTMF generation is a composite audio signal consisting of two tones, between the frequencies 697Hz and 1633Hz. The DTMF keypad is arranged such that each row will have its own unique tone frequency and also each column will have its own unique tone. The rows form the low tone group with frequencies between 697 Hz and 941 Hz. Columns form the high tone group with frequencies between 1209 Hz and 1633 Hz.

The tone frequencies are selected such that harmonics and intermodulation products will not cause an unreliable signal. No frequency is a multiple of another, the difference between any two frequencies does not equal any of the frequencies, and the sum of any two frequencies does not equal any of the frequencies.

Below is a representation of the typical DTMF keypad and the associated row and column frequencies.

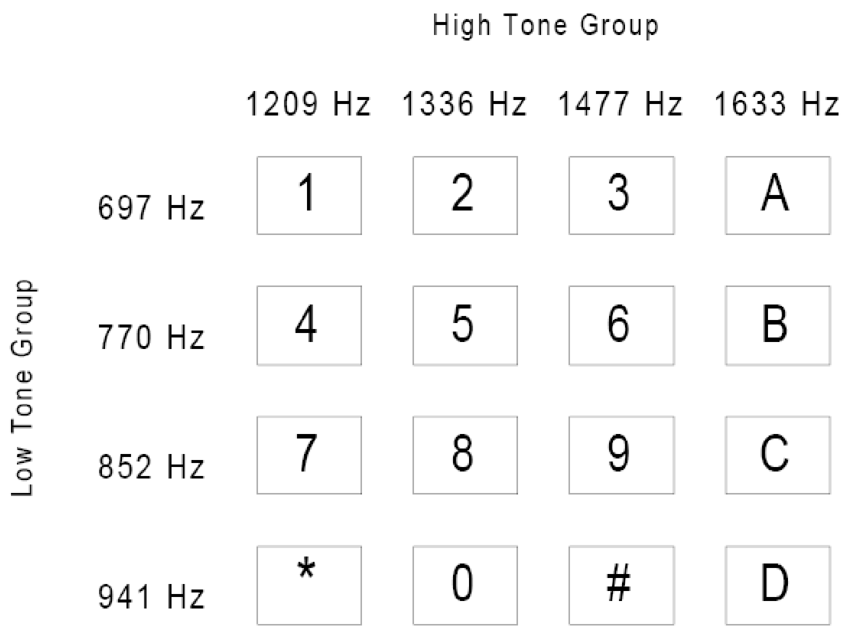
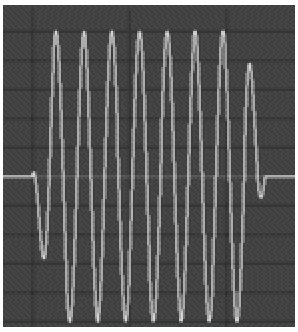


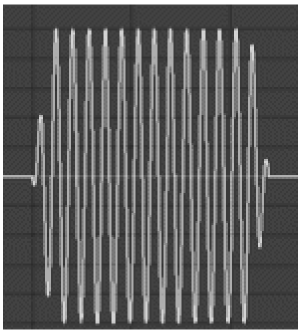
Fig. 5.1: Typical DTMF keypad and associated Row and Column Frequencies

As can be seen, each number is represented by a pair of tones. For example, in order to generate the DTMF tone for "1", we mix a pure 697 Hz signal with a pure 1209 Hz signal, as illustrated below.



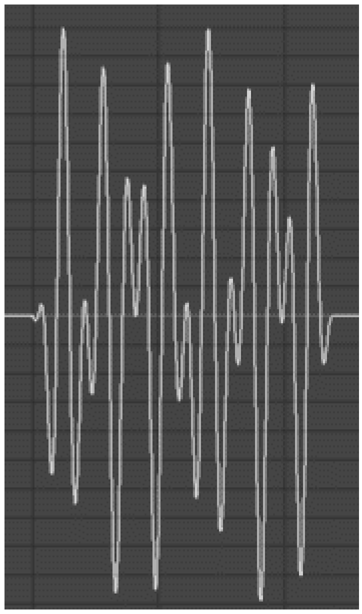
Pure 697 Hz tone

+



Pure 1209 Hz tone

=



DTMF tone "1"

Fig. 5.2: Generation of composite DTMF tone from individual tones

Signal Frequencies	Low Group	697, 770, 852, 941 Hz
	High Group	1209, 1336, 1477, 1633 Hz
Frequency Tolerance	Operation	<= 1.5%
	Non-Operation	>= 3.5%
Signal Duration	Operation	40ms max
	Non-Operation	23ms min
Signal Interruption		10ms max
Twist	Normal	8db max
	Reverse	4db max

Table 5.1 CCITT DTMF Recommendations

Let us see what the each of the recommendations mean. Any frequency within 1.5% of the DTMF frequency should be detected. Frequencies with 3.5% error should never be detected. Inside the 1.5% - 3.5% range is a don't care. DTMF signals lasting 40ms should

always be detected. Signals less than 23ms should never be detected. Inside the 23ms-40ms range is a don't care. DTMF signals that are interrupted for 10ms or less should not detect two separate signals. Twist is caused by a non-uniform power loss across the frequency spectrum. Normal twist is when low frequency power is greater than high frequency. Reverse twist is obviously the reverse condition. The detector must be reject 8db and 4db for normal and reverse twist respectively.

5.2 The UM91214B Tone/Pulse Dialer (DTMF Encoder)

The UM91214B is a single-chip, silicon gate, CMOS integrated circuit with an on-chip oscillator for a 3.58 MHz crystal or ceramic resonator. It provides dialing pulse or dual tone multi frequency (DTMF) dialing. A standard 4×4 matrix keyboard can be used to support either DP or DTMF modes. Upto 32 digits can be stored in the on chip RAM for redialing. In DTMF mode, minimum tone duration and minimum intertone pause support provide for rapid dialing. Maximum tone duration is dependent upon the key depression time in manual dialing.

Features

- Wide Operation voltage range: 2V to 5.5 V
- Uses inexpensive 3.58 MHz crystal
- Low standby current
- Built in Power up reset circuit
- One touch redial facility
- 32 digit capacity for redialing
- DTMF timing:
    Manual dialing: minimum duration for bursts and pauses
    Redial: calibrated timing

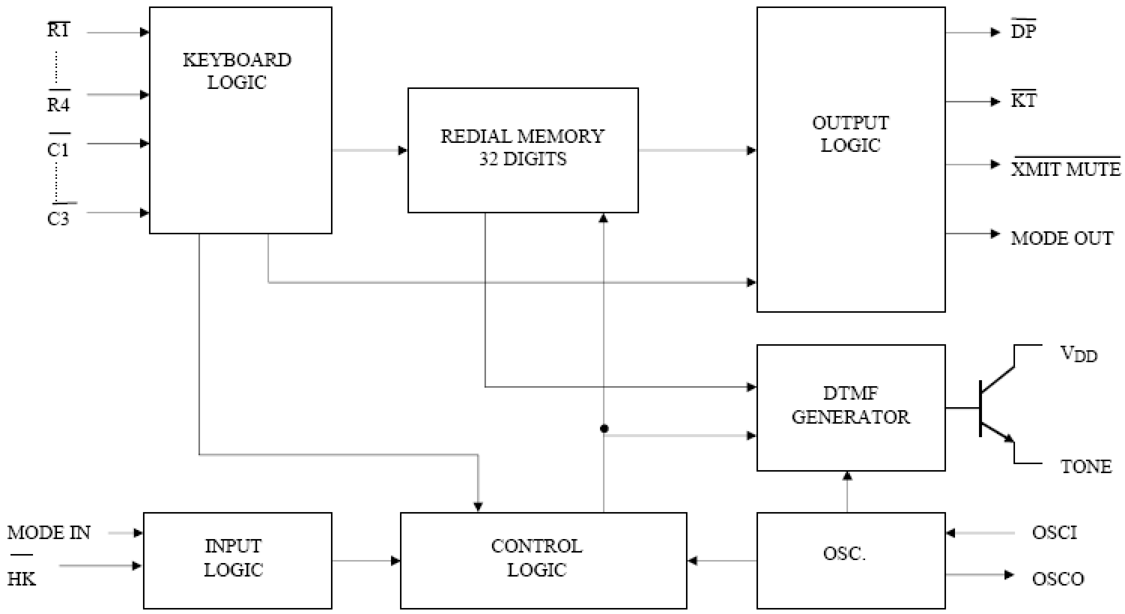


Fig. 5.3: Block diagram of UM214B

5.3 The MT 8870 DTMF Receiver

The MT 8870 is a full DTMF receiver that integrates both bandsplit filter and decoder functions into a single 18-pin DIP package. Manufactured using state-of-the art CMOS technology, the MT 8870 offers low power consumption (35 mW max) and precise data handling. It uses two 9th order bandpass filters with switched capacitors. These produce nice clean sine waves even from distorted inputs, with any harmonics suppressed. Its decoder uses digital counting techniques to detect and decode all 16 DTMF tone pairs into a 4-bit code. External component count is minimized by provision of an on-chip differential input amplifier, clock generator, and latched tri-state interface bus. Minimal external components required include a low cost 3.579545 MHz colour burst crystal, a timing resistor, and a timing capacitor.

Features

- Low power consumption
- Adjustable acquisition and release times
- Single 5 volt power supply
- Uses inexpensive 3.58 MHz crystal
- Dial tone suppression

Applications

- Telephone switch equipment
- Mobile Radio
- Remote Data Entry
- Remote Control

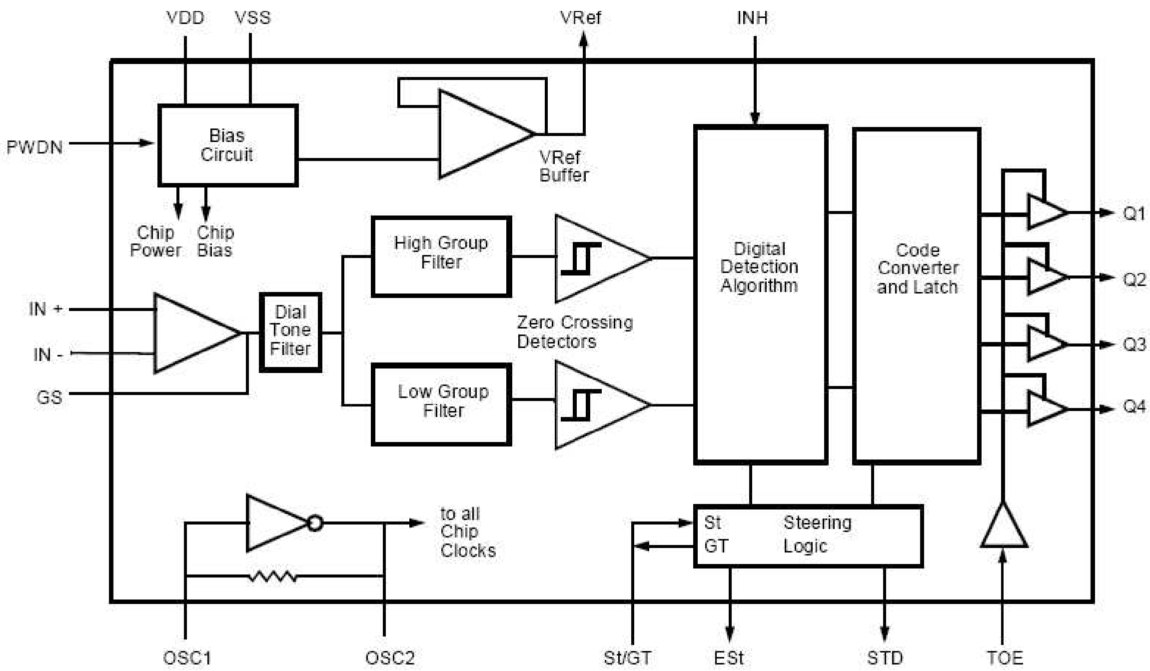


Fig. 5.4: MT8870 Functional Block Diagram

5.4 IC 74LS154 4-Line to 16-Line Decoder/Demultiplexer

The 74LS154 4-line-to-16-line decoder utilizes TTL circuitry to decode four binary coded inputs into one of sixteen mutually exclusive outputs, when both the strobe inputs, G1 and G2, are LOW. The demultiplexing function is performed by using the 4 input lines to address the output line, passing data from one of the strobe inputs with the other strobe input LOW. When either strobe input is HIGH, all outputs are HIGH. All inputs are buffered and input clamping diodes are provided to minimize transmission- line effects and thereby simplify system design.

Features

- Decodes 4 binary-coded inputs into one of 16 mutually exclusive outputs

- Performs the demultiplexing function by distributing data from one input line to any one of 16 outputs
- Input clamping diodes simplify system design
- High fan-out, low-impedance, totem-pole outputs
- Typical power dissipation 45 mW

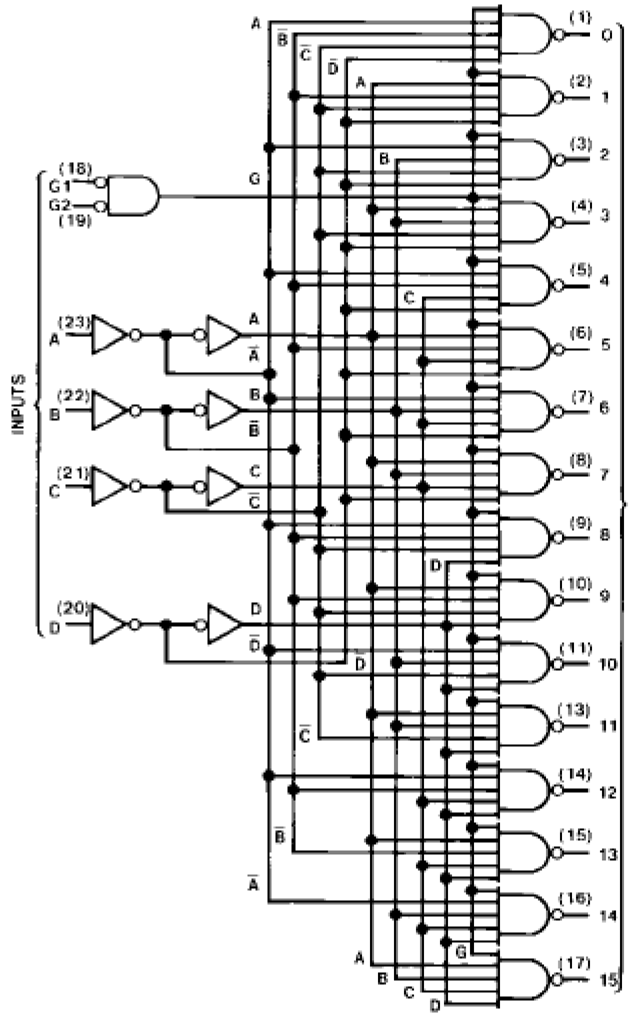


Fig. 5.5: IC 74LS154 Logic Diagram

5.5 IBM-PC Parallel Printer Port

IBM originally supplied three adapters that included a parallel printer port for its PC/XT/AT range of microcomputers. Depending on which were installed, each available parallel port's base address in the processor's I/O space would be one of 278 H, 378 H and 3BC H. Most contemporary PCs, shipped with a single parallel printer port have the base address at 378 H.

The PC parallel port adapter is specifically designed to attach printers with a parallel port interface, but it can be used as a general input/output port for any device or application that matches its input/output capabilities.

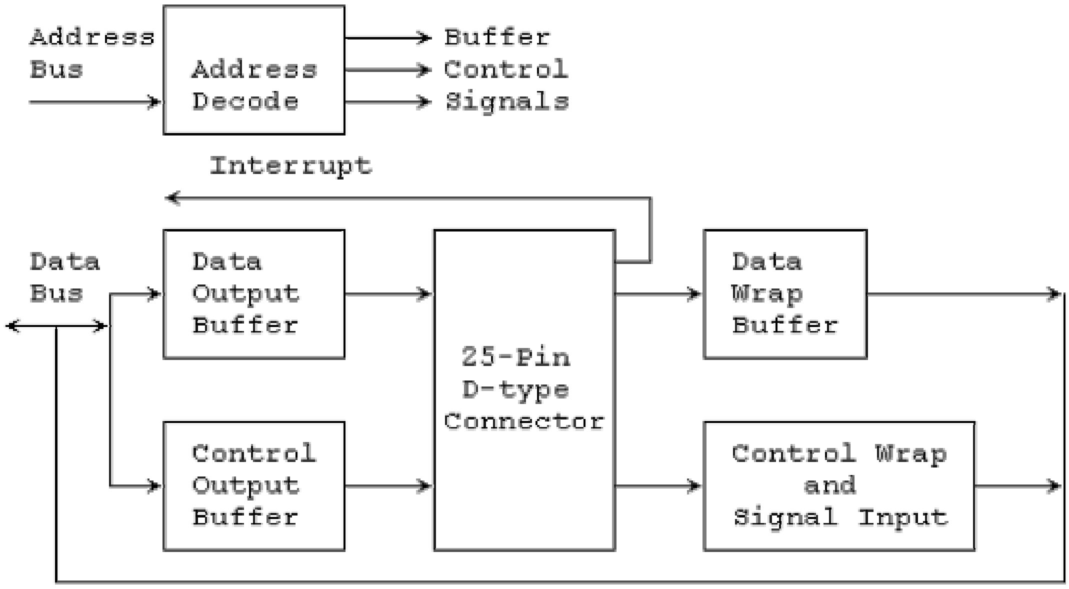


Fig. 5.6: Parallel Printer Adapter (IBM PC/AT) Block Diagram

The parallel port, as implemented on the PC, consists of a connector with 17 signal lines and 8 ground lines. The input/output signals are made available at the back of the adapter through a right-angled, PCB-mounted, 25-pin, D-type female connector. The signal lines are divided into three groups:

- Control (4 lines)
- Status (5 lines)
- Data (8 lines)

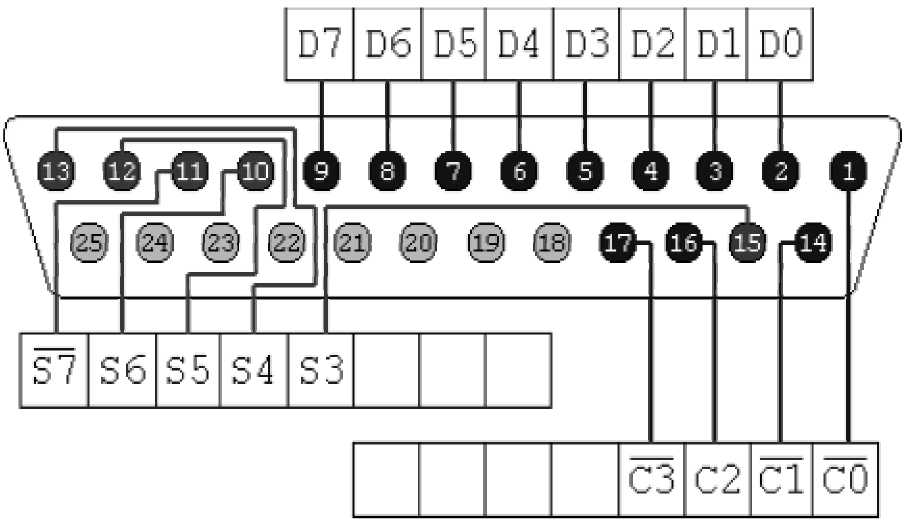


Fig. 5.7: The parallel port pins

As originally designed, the Control lines are used as interface control and handshaking signals from the PC to the printer. The Status lines are used for handshake signals and as status indicators for such things as paper empty, busy indication and interface or peripheral errors. The data lines are used to provide data from the PC to the printer, in that direction only. Later implementations of the parallel port allowed for data to be driven from the peripheral to the PC.

5.6 Microsoft Visual Basic

Visual Basic (abbreviated as *VB*) is Microsoft's high-level, object-oriented, rapid application development environment for the Windows platform. The first versions of Visual Basic were intended to target Windows 3.0 (a version for DOS existed as well), however it was not until version 3.0 for Windows 3.1 that this programming language gained large-scale acceptance in the shareware and corporate programming community.

Using drawing tools that resemble those found in hardcopy page layout programs or PhotoShop, VB programmers make user interfaces by drawing controls and other UI components onto forms. The programmer then adds code to respond to user interactions with the controls (for example, clicks, drag and drop, etc) known as events. The code can trigger events in other controls (for example, by displaying text or an image), execute procedures (run some algorithm based on the values entered in some control, output data, do business logic, etc), or almost anything else one might do in code.

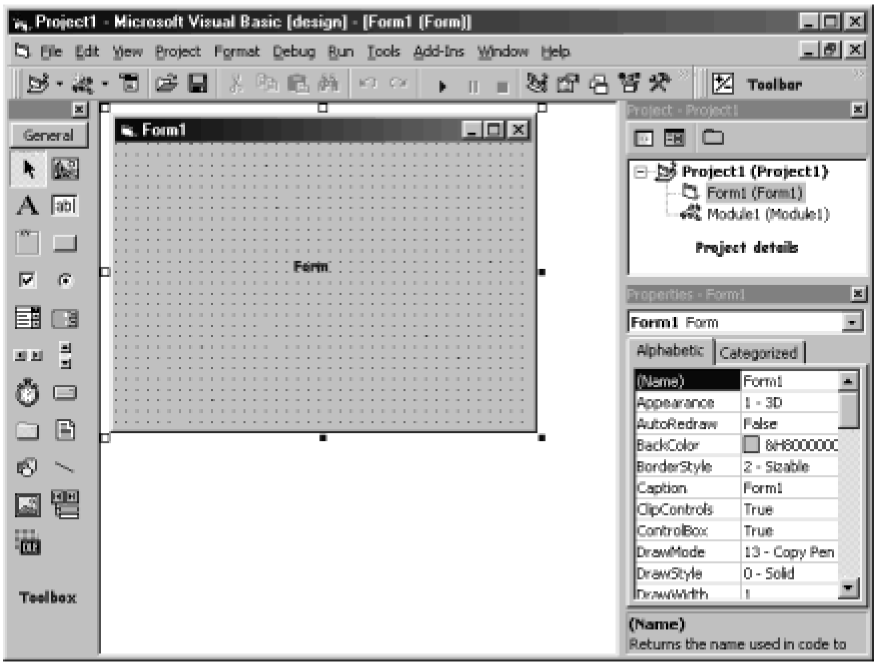


Fig. 5.8: Screenshot of Visual Basic Main Window

Visual Basic is an interpreted language like its ancestor BASIC, with appropriate modifications to accommodate object-oriented programming, and is weakly typed. That is, the VB development environment goes to great lengths to format (and aid the user in formatting) programming code so that it conforms to executable syntax. For example, VB will appropriately change the case of newly typed variable names to match those that have been declared previously (if they have been declared at all!).

Traditionally, VB is known for compiling programs into pseudo-code which is interpreted at runtime, requiring the use of dynamically-linked libraries. However, newer versions can compile code into something more closely resembling the efficient machine code generated by C-like compilers.

For new Windows programmers, VB offers the advantage of being able to access much of the Windows UI functionality without knowing much about how it works by obscuring the technical details. Although accessing low-level Windows UI functionality is possible, doing so in VB is as, or more difficult compared to such access using Visual C++ or other lower level programming languages. The language is garbage collected, has a large library of utility objects, and has basic object oriented support. Unlike many other programming languages, Visual Basic is not case sensitive. String comparisons can be performed with case sensitivity, if so desired.

Visual Basic was designed to be usable by all programmers, whether novice or expert. Forms are created using drag and drop techniques. A tools palette is used to place controls (e.g., text boxes, buttons, etc.) on the form (window). Controls have attributes and event handlers associated with them. Default values are provided when the control is created, but may be changed by the programmer. Many attribute values can be modified during run time based on user actions or changes in the environment, providing a dynamic application.

A Visual Basic application can consist of one or more windows, or a single window that contains child windows, as provided by the operating system. Dialog boxes with less functionality (e.g., no maximize/minimize control) can be used to provide pop-up capabilities. Controls provide the basic functionality of the application, while programmers can insert additional logic within the appropriate event handlers. For example, a drop-down combination box will automatically display its list and allow the

user to select any element. An event handler is called when an item is selected, which can then execute additional code created by the programmer to perform some action based on which element was selected.

Alternatively, a Visual Basic component can have no user interface, but be available to other programs, providing objects that implement functionality. This allows for server-side processing or an add-in model.

Using custom controls provided by Microsoft or third parties, almost any functionality that is possible in Windows can be added to a VB program by drawing a custom control onto a form in the project. Visual Basic spawned the first commercially viable reusable component market. There are thousands of third party components available today from hundreds of vendors. Visual Basic makes it easy to build, deploy, use, and reuse components; however it is not as easy to use forms created for one application with another, due to the global nature of the language.

5.7 Inpout32.dll for Parallel Port Interfacing

Visual basic is a very fast and easy tool for developing applications with high degree of user friendliness, but it lacks some important functionality such as direct access to hardware. The easy and effective solution for this problem is to use a DLL. One such DLL is the inpout32.dll.

This DLL is a very useful for interfacing parallel/serial ports. Since Visual Basic does not have any facility to communicate with parallel port directly, programmers will have to write code for accessing parallel port in a DLL and should be called from VB. In this DLL, accessing the device registers is done using functions _Inp() and _Outp(), which is declared in conio.h. More details about these functions can be found at MSDN online.

We will export two functions from the DLL. One is for reading from a device register and the other is for writing to the device register. Let these functions be InPort() and OutPort(). InPort() will take one parameter i.e. the address of the register to be read and Outport will take two i.e. the address of the register to which the data to be written and the data itself.

The outstanding feature of Inpout32.dll is that it can work with all the windows versions without any modification in user code or the DLL itself. Let us see how it is achieved, what programming methods used, what are the APIs used, etc. The dll will check the operating system version when functions are called, and if the operating system is WIN9X, the DLL will use _inp() and _outp functions for reading/writing the parallel port. On the other hand, if the operating system is WIN NT, 2000 or XP, it will install a kernel mode driver and talk to parallel port through that driver. The user code will not be aware of the OS version on which it is running. This DLL can be used in WIN NT clone operating systems as if it is WIN9X. The flow chart of the program is given below.

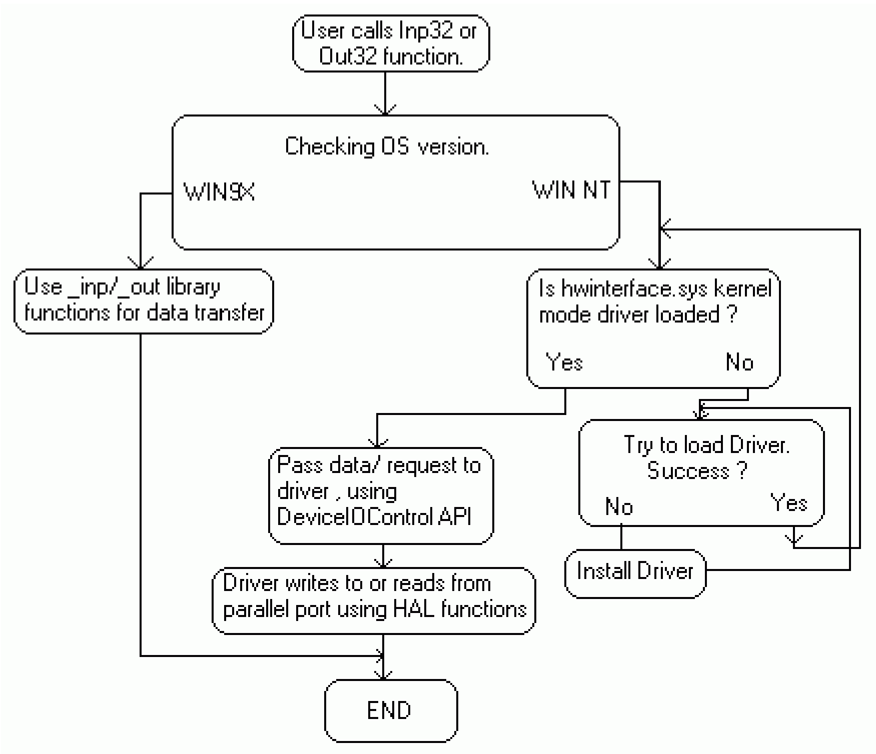


Fig. 5.9: inpout32.dll Flow Chart

5.8 DC Motors

The classic DC motor has a rotating armature in the form of an electromagnet with two poles. A rotary switch called a commutator reverses the direction of the electric current twice every cycle, to flow through the armature so that the poles of the electromagnet push and pull against the permanent magnets on the outside of the motor. As the poles of the armature electromagnet pass the poles of the permanent magnets, the commutator reverses the polarity of the armature electromagnet. During that instant of switching polarity, inertia keeps the classical motor going in the proper direction.

DC motor speed generally depends on a combination of the voltage and current flowing in the motor coils and the motor load or braking torque. The speed of the motor is proportional to the voltage, and the torque is proportional to the current. The speed is typically controlled by altering the voltage or current flow by using taps in the motor windings or by having a variable voltage supply.

5.8.1 Working of a DC Motor

DC motors consist of rotor-mounted windings (armature) and stationary windings (field poles). In all DC motors, except permanent magnet motors, current must be conducted to the armature windings by passing current through carbon brushes that slide over a set of copper surfaces called a *commutator*, which is mounted on the *rotor*. The commutator bars are soldered to armature coils. The brush/commutator combination makes a sliding switch that energizes particular portions of the armature, based on the position of the rotor. This process creates north and south magnetic poles on the rotor that are attracted to or repelled by north and south poles on the *stator*, which are formed by passing direct current through the field windings. It is this magnetic attraction and repulsion that causes the rotor to rotate.

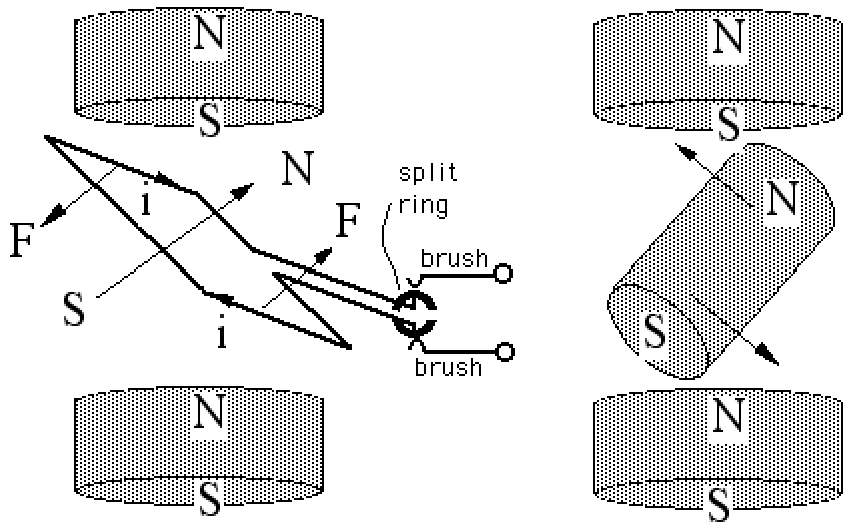


Fig. 5.10: DC Motor working

The motor rotates toward the pole alignment point. Just as the motor would get to this point, the brushes jump across a gap in the stator rings. Momentum carries the motor forward over this gap. When the brushes get to the other side of the gap, they contact the stator rings again and the polarity of the voltage is reversed in this set of rings. The motor

begins accelerating again, this time trying to get to the opposite set of poles. (The momentum has carried the motor past the original pole alignment point.) This continues as the motor rotates.

5.8.2 Advantages of DC Motors

5.8.2.1 Easy to understand design

The design of the brushed DC motor is quite simple. A permanent magnetic field is created in the stator by either of two means:

- Permanent magnets
- Electro-magnetic windings

If the field is created by permanent magnets, the motor is said to be a "permanent magnet DC motor" (PMDC). If created by electromagnetic windings, the motor is often said to be a "shunt wound DC motor" (SWDC). Today, because of cost-effectiveness and reliability, the PMDC motor is the motor of choice for applications involving fractional horsepower DC motors, as well as most applications up to about three horsepower.

At five horsepower and greater, various forms of the shunt wound DC motor are most commonly used. This is because the electromagnetic windings are more cost effective than permanent magnets in this power range. In most DC motors, several sets of windings or permanent magnets are present to smooth out the motion.

5.8.2.2 Easy to control speed

The greatest advantage of DC motors may be speed control. Since speed is directly proportional to armature voltage and inversely proportional to the magnetic flux produced by the poles, adjusting the armature voltage and/or the field current will change the rotor speed. The higher the armature voltage, the faster the rotation. This relationship is linear to the motor's maximum speed.

5.8.2.3 Easy to control torque

In a brushed DC motor, torque control is also simple, since output torque is proportional to current. If you limit the current, you have just limited the torque which the motor can achieve. This makes this motor ideal for delicate applications such as textile manufacturing and for mobile applications.

5.8.2.4 Simple, cheap drive design

The result of this design is that variable speed or variable torque electronics are easy to design and manufacture. Varying the speed of a brushed DC motor requires little more than a large enough potentiometer. However in practice, these have been mostly replaced by the SCR and PWM drives, which offer relatively precise control over voltage and current.

5.8.2.5 Back emf

Here is an interesting corollary. Every motor is a generator. The emf that it generates is called the *back emf*. The back emf increases with the speed, because of Faraday's law. So, if the motor has no load, it turns very quickly and speeds up until the back emf, plus losses, equals the supply voltage. The back emf can be thought of as a 'regulator', if you like: it stops the motor turning too quickly. When the motor is loaded, then the phase of the voltage becomes closer to that of the current (it starts to look resistive) and this apparent resistance gives a voltage. So the back emf required is smaller, and the motor turns more slowly.

After more than a century, DC motors are still in widespread use, and thanks to the numerous applications that show no signs of disappearing, they will be around for many years to come.

5.9 DC Motor Controller - The H-Bridge Circuit

The motor controller circuit, known as the H-Bridge circuit, is capable of controlling the direction of a dc motor - forward, reverse or stop. The schematic of an H-Bridge circuit is shown below.

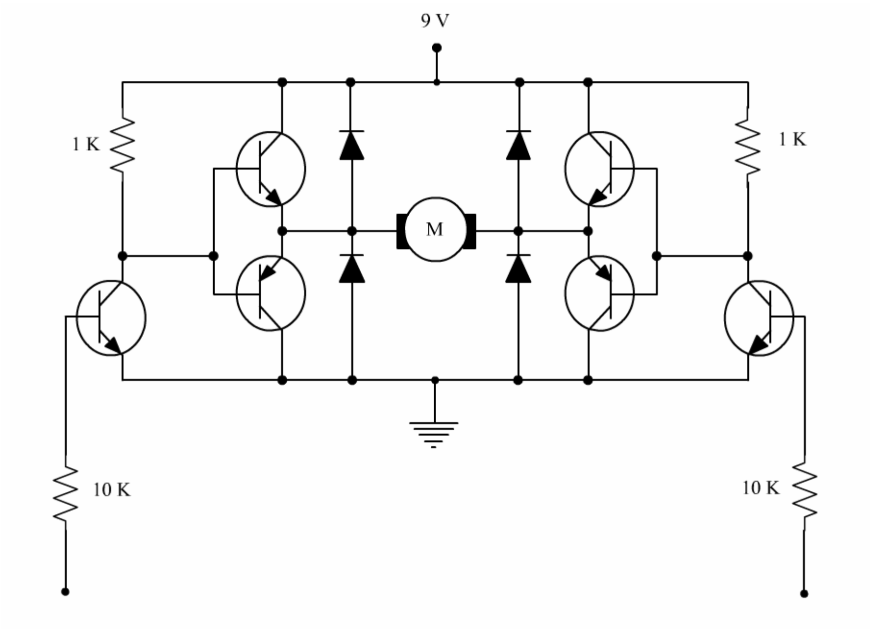


Fig. 5.11: H-Bridge Schematic

When both the points 'A' and 'B' are HIGH, transistors Q1 and Q2 are in saturation. Hence the bases of Q3 to Q6 are grounded. Hence Q3 & Q5 are OFF and Q4 & Q6 are ON. The voltages at both the motor terminals are the same and hence the motor is OFF. Similarly when both A and B are LOW the motor is OFF.

When A is HIGH and B is LOW, Q1 saturates, Q2 is OFF. The bases of Q3 and Q4 are grounded and those of Q4 and Q5 are HIGH. Hence Q4 and Q5 conduct, making the right terminal of the motor more positive than the left, and the motor turns ON in one direction. When A is LOW and B is HIGH, the left terminal of the motor is more positive than the right and the motor rotates in the reverse direction. The diodes protect the transistors from the surge produced due to the sudden reversal of the motor. The logic for the H-Bridge is given below.

Inputs		Motor Status
A	B	
0	0	OFF
0	1	ON (Clockwise)
1	0	ON (Anti Clockwise)
1	1	OFF

Table 5.2: The H-Bridge Motor Controller Logic

5.10 Optocouplers

There are many situations where signals and data need to be transferred from one subsystem to another within a piece of electronics equipment, or from one piece of equipment to another, without making a direct ohmic electrical connection. Often this is because the source and destination are (or may be at times) at very different voltage levels, like a microprocessor which is operating from 5V DC but being used to control a triac which is switching 240V AC.

In such situations the link between the two must be an isolated one, to protect the microprocessor from overvoltage damage. Relays can of course provide this kind of isolation, but even small relays tend to be fairly bulky compared with ICs and many of today's other miniature circuit components. Because they are electro-mechanical, relays are also not as reliable and only capable of relatively low speed operation.

Where small size, higher speed and greater reliability are important, a much better alternative is to use an optocoupler. These use a beam of light to transmit the signals or data across an electrical barrier, and achieve excellent isolation. Optocouplers are essentially a combination of two distinct devices: an optical transmitter, typically a gallium arsenide LED (light-emitting diode) and an optical receiver such as a phototransistor or light-triggered diac. The two are separated by a transparent barrier which blocks any electrical current flow between the two, but does allow the passage of light.

Optocouplers are essentially digital or switching devices, so they're best for transferring either on-off control signals or digital data. Analog signals can be transferred by means of frequency or pulse-width modulation.

The MCT2E optocoupler consists of a gallium-arsenide infrared-emitting diode and an n-p-n silicon phototransistor. The pin diagram and internal diagram of the MCT2E Optocoupler is shown below.

Fig. 5.12: MCT2E Pin Diagram and internal diagram

Chapter 6

APPLICATIONS

iBot is a video surveillance vehicle, hence its main and intended application is remote surveillance. However, due to its flexible design it can be used for a large variety of applications with only minor modifications. It can be used in various industries, organizations and even homes. A few specific uses from the limitless possibilities are

- Can be used as a security device in homes to prevent burglaries and intrusions.
- Can be used in factories to keep track of worker activities.
- Can be used in law enforcement for reconnaissance to avoid human casualties.
- Can be used in search and rescue operations when it would be a potentially dangerous situation for humans to attempt.
- Can be used to probe into hard to reach places such as caves or mines. Hence it can be used in resource mapping and related operations.
- Can be used for surveying hazardous environments.

Chapter 7

SCOPE FOR MODIFICATION

iBot is a very versatile device. Since it was built from the beginning with a strictly modular design process, it presents scope for improvement without any major changes in hardware or software. Some suggestions for improvement include

- More robust design to enable it to operate even in harsh conditions
- Increasing the range of operation by using more powerful transmitters
- Improving response time and decreasing data transmission overhead by implementing a microcontroller on the robot itself
- Adding a complex neural network to give it a large degree of artificial intelligence
- Implementing additional features such as live telemetry
- Implementing collision detection and avoidance by performing image processing instead of using sensors

Chapter 8

CONCLUSION

It can be seen that *iBot* is a very versatile device. It is a pioneering project encompassing both software and hardware Apart from surveillance, the robot can be used for reconnaissance, rescue, tracking etc. It can be operated in either manual or automatic modes, as desired by the user. It encompasses all essential features of a security device. It has a modular design and provides scope for improvement without much change in hardware. In all, *iBot* seems to be a competent surveillance device for the future.

REFERENCES

[01] Al's Robotics [Online] http://www.alsrobotics.co.uk

[02] DialABC [Online] http://www.dialabc.com/sound/dtmf.html

[03] Discover Circuits [Online] http://www.discovercircuits.com

[04] E.J. Bantz. *Programming Video for Windows* [Online] http://ej.bantz.com/video/detail/

[05] Evangelos Petroutsos. *Mastering Visual Basic 6*. ISBN: 0782122728

[06] Jan Axelson. *Parallel Port Complete*. ISBN: 8187972025

[07] Microsoft Encarta Online Encyclopedia [Online] http://encarta.msn.com

[08] P. A. Kulkarni, S. G. Karad. *Robotics And Remotely Operated Vehicles* [Online] http://www.isprs.org/istanbul2004/comm5/papers/677.pdf

[09] Robotics India [Online] http://www.roboticsindia.com

[10] Seattle Robotics [Online] http://www.seattlerobotics.org

[11] Tomi Engdahl. *Parallel Port Interfacing Made Easy* [Online] http://www.epanorama.net/circuits/parallel_output.html

[12] Vikas Patial. *F.O.B.O - Floppy Box Robot*. Meerut Institute of Engineering and Technology.

[13] Wade Tyler Carpenter et al. *Wireless Roaming Web Cam* [Online] http://www.ece.ubc.ca/~elec474/reports/dec01/group2.pdf

[14] Wikipedia - The Free Encyclopedia [Online] http://en.wikipedia.org